

CLAIMS:

1. A software architecture comprising a plurality of modules (M0-M4), at least one module of said plurality being a module (M1) adapted to call another (M3) of said plurality of modules using a reference (&M3) to said called module,

wherein the reference (&M3) of the module to be called is supplied as an input to said calling module (M1).

2. A software architecture as claimed in claim 1, wherein each of said plurality of modules (M0-M4) is adapted to recognize as a null reference an input parameter having a predetermined value and to not make a call when the module to be called is indicated by the null reference.

3. A software architecture according to claim 1, wherein each module (Mx) corresponds to a software entity selected in the group consisting of functions, procedures, operating system tasks, and layers.

4. A method of producing a new module-based software architecture based on an existing module-based architecture comprising a plurality of modules (M0-M4), at least one module of said plurality being a module (M1) adapted to call another (M3) of said plurality of modules using a reference (&M3) to said called module, wherein the reference (&M3) of the module to be called is supplied as an input to said calling module (M1), the method comprising the steps of:

removing at least one of said plurality of modules (M3), and

altering the value of inputs corresponding to the reference (&M3) of the removed module.

5. An architecture-producing method according to claim 4, wherein each of said plurality of modules (M0-M4) is adapted to recognize as a null reference an input parameter having a predetermined value and to not make a call when the module to be called is

indicated by the null reference, and wherein the altering step comprises replacing inputs corresponding to the reference (&M3) of the removed module with a null reference.

6. An architecture-producing method according to claim 4, and comprising the step of replacing the removed module by a replacement module (M5) having a different reference (&M5), wherein the altering step comprises replacing inputs corresponding to the reference (&M3) of the removed module with inputs corresponding to the reference (&M5) of the replacement module.

7. An architecture-producing method according to claim 4, wherein each module (Mx) corresponds to a software entity selected in the group consisting of functions, procedures, operating system tasks, and layers.

8. A radio telephone including a phase locked loop intended to be controlled by means of a radio driver software having an architecture according to claim 1.